Apr 23rd, 1:00 PM

Paper Session III-A - Artificial Expertise in Systems Engineering

Daniela T. Iorgluescu
Martin Marietta Astronautics Group

Carol S. Giffen
Martin Marietta Astronautics Group

David F. Giere
Martin Marietta Astronautics Group

ARTIFICIAL EXPERTISE IN SYSTEMS ENGINEERING

DANIELA T. IORGULESCU*, DAVID F. GIERE**, CAROL S. GIFFEN***

MARTIN MARIETTA ASTRONAUTICS GROUP
POST OFFICE BOX 179
DENVER, COLORADO 80201

ABSTRACT

As technology development and engineering problems have grown in complexity, technical systems have evolved to meet these challenges. This evolution has occurred within a foundation of traditional engineering analysis and work processes originating prior to current computer technology. These processes were designed to improvise and compensate for ambiguous design or analysis information. Systems engineering optimization of computer technology applications can eliminate or redesign engineering processes such that the unified system function focuses on innovation, flexibility, speed, and quality. "Artificial Expertise" for systems engineering refers to the application of artificial intelligence expert systems and shared data bases to promote the integration of cross-functional engineering groups through technical interchange and control mechanisms. This paper presents some conceptual applications and examples for implementing artificial expertise in system development.

SYSTEMS ENGINEERING

Aerospace systems engineering technology is composed of functional methodologies and engineering skills applied toward complex system development. Mission objectives are transformed into total system definition and design. The function of assuring system compatibility among the diverse system elements is achieved through complete system integration of design requirements and functional interfaces. These occur within a dynamic framework of variant hardware, software, schedule, cost, analysis, and support requirements. As systems have evolved and matured, the complexity of these dynamic interactions has increased and presented challenges to the systems engineering coupling of scientific and management techniques.

From the concept exploration phase to the design phase of the system development cycle, systems engineering provides the iterative process to assure requirement definition, system integration, and requirement verification.

REQUIREMENTS DEFINITION

Requirements definition provides a formalized method for translating mission objectives into quantifiable scientific descriptions that provide the measure for judgement of design solutions. These requirements begin at the overall system level and are gradually decomposed to a level of sufficient detail for hardware or operational software implementation. Many low level component requirements can only be defined during the design cycle as a function of analysis iterations and system evolution. As the requirements documents mature, their requirements allocation takes the form of a specification tree based on system hierarchy. The requirements hierarchy starts at the system level and is decomposed into finer detail at the segment, element, subsystem, and component levels. The system engineering approach involves stating the mission objectives and identifying ambiguous areas that will be defined during design evolution.

SYSTEM INTEGRATION

System integration identifies the inter-relationships among the technical disciplines and the requirements allocation that couple to form the total system definition. The product of this iterative effort is a unified design such that independent software, hardware, or operational entities will be well integrated into the final system.

To illustrate this concept, a simplified example of a payload integration design cycle is shown in Figure 1. Mission objectives of

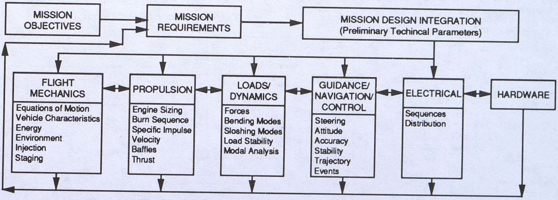

Figure 1. Simplified Payload Integration Design Cycle

* Senior Flight Software Engineer, B.S. in Mechanical Engineering, M.S. in Special Studies (Electrical Engineering), Member AIAA
** Director, Technical Operations, B.S. in Aerospace Engineering, M.S. in Aerospace Engineering, Member AIAA
*** Manager, Systems Engineering Service Orders, B.A. in Biology

delivering a payload to low earth orbit are transformed into requirements for specific orbital parameters. These are now technical parameters in the mission design. They are implemented by Flight Mechanics in a three or six degree of freedom trajectory simulation providing performance sizing and optimization. Next, Propulsion provides the desired thrust energy to achieve the required flight profile from Flight Mechanics. Propulsion thrust impacts the Guidance/Navigation/Control attitude control analysis and the Loads/Dynamics bending moments resulting from the thrust. All of these analyses are further constrained by hardware limitations which may redefine mission requirements, thus, providing the potential for design reiteration. System integration focuses on interfaces and compatibility to assure that mission requirements will be met.

REQUIREMENT VERIFICATION

Systems engineering verifies compliance of mission requirements with the analysis or testing results. Each requirement is tested for proper implementation using standard verification methodology (test, analysis, inspection). Requirements traceability is important for assuring that a specific requirement which has been allocated and verified, properly reflects upper level requirements from which it is derived. Also, traceability assists in assuring that all system requirements are accounted for prior to allocation.

CONCEPTS OF QUALITY AND CONCURRENT SYSTEMS

The driver for product/process improvement initiatives is the desire to optimize quality. The degree to which quality is not achieved is reflected in detrimental cost and schedule impacts. Although system inspections, tests, and audits attempt to improve quality by aiming to discover defects and fixing them, the approach to attaining quality is prevention. "Do it right the first time" implies zero defects and requires an explicit measurement of quality defined as the Cost of Quality. The Cost of Quality is the sum of the Price of Nonconformance (the price for doing the job incorrectly) plus the Price of Conformance (the price paid for making certain that requirements are met the first time).[1] Examples of the Price of Nonconformance include change orders, testing reruns, rework, redesign, downtime, and revisions. Management techniques are developed to minimize this effect.

The Total Quality Management initiative incorporates the principles of concurrent engineering. Concurrent engineering is "a systematic approach to the integrated concurrent design for products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of product life cycle from conception through disposal including quality, cost, schedule, and user requirements".[2] Implementation of concurrent engineering requires a departure from traditional design which is sequential in nature to a dynamic real-time iterative interaction between the design phases. Concurrent box structure design techniques replace sequential programs by concurrent execution of distributed processing environments exploiting software and hardware parallel processing.[3]

Concurrent engineering implies changes to information processing within the organization. Techniques are needed to quickly process the sea of data generated. Managing this knowledge can serve as a strategic corporate resource. A company's competitive edge is a function of the flexibility and rapidity of information synthesis and incorporation into downstream processes. Expert systems provide an avenue for information synthesis through transaction processing and high level decision support. They have the potential of producing order from chaos and enabling concurrent engineering processes.

ARTIFICIAL EXPERTISE

The inseption of the artificial intelligence discipline occurred in the late 1950's and focused on exploiting the problem solving techniques of human beings to solve problems.[4] Rapid advances in computer technology combined with a scarcity of knowledge about how the brain works biologically contributed to its rapid development. During the last seven years, advances in neuroscience and computers have fueled the parallel discipline of neural networks. It considers a different approach to problem solving – that of replicating how the brain works biologically. Branches of artificial intelligence technology include natural language processing, visualization systems, automatic programming, robotics, and expert systems.

EXPERT SYSTEMS TECHNOLOGY

Expert systems are software programs designed to capture knowledge assimilated by an expert, group of experts, or information sources such as books or manuals. Expert system design seeks to emulate human thinking by embodying the way an expert represents, utilizes, and acquires that knowledge. This emulation includes memory organization, processing limitations, and reasoning strategies. Expert thinking is studied within the framework of human information processing: feature detection, part segmentation, memory template match, and best match selection.[5] An expert system has three basic parts as illustrated in Figure 2:

- a knowledge base of facts and heuristics (rules) associated with the problem;
- an inference engine for applying the knowledge base including rule interpretation for the problem solution; and
- a dynamic global data base serving as a working memory of the problem status and input data.

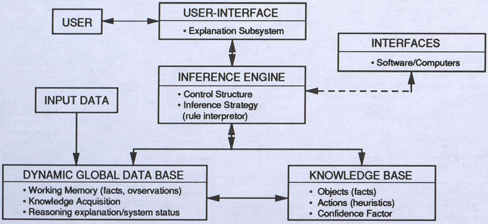

Figure 2. Basic Expert Systems Components

Expert systems are different from data processing programs because of their extensive use of heuristics, symbolic manipulation, and inferencing mechanisms to represent and extract knowledge from a large knowledge base.[6] Data processing programs access large data bases by well defined, deterministic, and repetitive algorithms. Expert systems apply heuristic (codified rules of thumb) search to control the process rather than random or exhaustive search techniques where every possible combination of rules and facts is tested to reach a conclusion. The difference lies in processing on reasoning-based information extraction rather than brute force number crunching to the computer's capacity. Expert systems differ from data bases in their ability to manage symbol/rule relationships to a higher level of abstraction. They employ a subtler form of thinking made possible through the use of rules of logic, extraction of human knowledge, and knowledge engineering concepts. The expert system approach assumes that the problem cannot be defined in detail but that heuristic solutions exist.

Knowledge Base. The knowledge base comprises the factual and heuristic, rule oriented, and structure oriented knowledge. The information stored in the knowledge base is like that contained in traditional data bases. The data base contains rules to manipulate its facts. Those rules that mimic human judgement are referred to as "heuristics". The knowledge base is composed of objects and actions.

The objects are computer symbols of physical or abstract concepts. Objects are linked together to form semantic networks with specific terms (IS-A, HAS-A, CAUSED BY, DEFINITION) defining their corresponding relationships. Objects and their relations may be stored in frames which are tabular data structures. Frames can be organized in a hierarchical collection of frames that inherit information from frames above them. The inheritance feature can serve as a form of inference in deductive logic employed in the inference engine. Object oriented programming activates procedures that are attached to objects when messages are received from other objects.

Actions modify the situation or the relevant data base. Actions are represented by rules that are expressed in IF-THEN format with ANDs and ORs for more complex relationships. Actions may be activated by messages (object oriented programming) or changes in the global data base. The degree to which knowledge is known to be correct or affecting final assertions is reflected in the confidence factor. This accommodates "fuzzy logic" for imprecise or incomplete knowledge. The coupling of frame based hierarchical systems and object oriented programming allow closer modeling to the real world through linkage of the knowledge representation system and the system's reasoning process; thus, reducing the amount of explicit knowledge needed by the expert system.[7]

Inference Engine. The inference engine provides the inference strategy and control structure so that the knowledge base can be used effectively. Expert system tools or shells have capitalized on the idea that inference engines are essentially knowledge independent and can be used on different knowledge bases. Inference is the process of developing new facts based on established facts. As computer programming allowed the manipulation of numerical symbols to solve equations in mathematical logic, so can non-numerical symbols (objects) be manipulated to program human deductive logic. The program uses rules in the general form of IF-THEN statements and performs logical operations by searching for and matching symbols or objects. For example, IF A is a member of class B, and IF members of B have property C, THEN it can be inferred that A has property C. By substituting values for the symbols, expert systems can make inferences through pattern matching between the knowledge base, inference rules, and input facts as follows:

Inference Engine:	IF A (premise) THEN B (hypothesis)
User Interface:	Input fact (A) = APOLLO is a workstation Output new fact (B) = APOLLO is a computer (inferred)
Knowledge Base:	IF APOLLO is a workstation THEN APOLLO is a computer

The control structure directs the reasoning process so that all possible combinations of rules and facts are not tested between the knowledge base and the inference engine. This is accomplished through search techniques known as forward and backward chaining. Forward chaining starts with the known facts while working toward an unknown solution by matching the premises from the rules that are verified by the facts. The conclusion of the rule then acts upon the knowledge by adding new facts. Backward chaining begins with a solution and gathers facts to support it by matching the hypotheses of the rules and then seeks to verify the premise of those rules by searching the knowledge base. Groupings of rules in the inference engine are classed by meta rules which prioritize the order in which rules are tried. The black board inference technique involves a group of cooperating expert systems that control solution development on a common data structure known as a "black board".[8]

Most expert systems are rule based systems handling empirical relationships that are described as shallow representations of knowledge. Collections of "IF A THEN B" rules represent this type of shallow knowledge. Deeper representations allow the system to restructure its knowledge, prioritize and break its own rules if it has to, and react to exceeding the bounds of its expertise. A great deal of research continues in this area.

EXPERT SYSTEMS IMPLEMENTATION

Expert systems provide the opportunity of fusing the knowledge of many experts and enriching the problem solving techniques to produce superior quality results. Through continuous incremental development of the knowledge base to reflect dynamic conditions, the organization has an opportunity to manage knowledge – its key technology. The expert system can suggest solutions to a problem and explain its reasoning by providing a record of the information and intermediate conclusions it used. Expert systems are designed to solve problems through reasoning rather than numerical techniques. Reasoning is achieved through symbolic processing by reaching conclusions from valid or invalid premises. The quality of the conclusions is a direct function of the fidelity of the knowledge base and crude in the sense that it does not substitute human judgement which reflects intuition, common sense, and emotion. Expert systems encapsulate knowledge that aids decision making within the organization by providing the framework for a structured decision process with reduced uncertainty through defined constraints. Quality is enhanced through the development of a standardized consistent approach to problem solving resulting in more wholistic evaluations.

Expert systems employ techniques which have demonstrated certain problems: interdependency of rules where rules conflict with each other, difficulty of solution near domain knowledge boundary, or problems due to multiple inferencing paradigms. The data driven expert system software is structurally and functionally different from conventional software that is procedural driven sequential algorithmic code. These differences indicate that normal test techniques cannot be used to validate expert systems. Research in this area continues.

APPLICATIONS IN AEROSPACE SYSTEMS

Artificial expertise concepts have been combined with human information processing to develop real-time expert systems providing a foundation for requirements definition, system integration, and requirement verification. Through knowledge and processing function descriptions in terms of abstraction hierarchy, a system state of knowledge and organization has been achieved for specific domains.

REQUIREMENTS DEFINITION

Data Bases. Decomposition of the hierarchical system requirements into implementable hardware/software requirements has been aided with highly versatile engineering data bases. The

Flight Telerobotic Services (FTS) functional requirements development were conducted with the aid of a software data base tool referred to as NASREM (NASA/NBS Standard Reference Model).[9] This provided generic hierarchical, functional control and interfaces. The System Engineering Data Base (SEDB) was developed by Martin Marietta as a requirements analysis tool to assist the development of systems engineering requirements. The data base is sufficiently versatile to maintain complete system requirements traceability (parent/children/siblings) along with relationships to verification methods, requirement sources, and design compliance. The SEDB provides configuration allocation management control of system requirements and is currently applied in production.

Expert Systems. Expert systems have been developed for mission planning systems and task flow requirements. The Launch Resource Scheduling System (LRS II)[10] was developed for use by USSPACECOM to assess launch capability to meet future satellite requirements. It uses multiple knowledge bases to match satellite launch requirements with available launch vehicles, upper stages, and launch pads in order to establish a launch manifest. A prototype scheduling system (MAESTRO)[11] was developed by Martin Marietta for spacecraft and their experiments. MAESTRO techniques include heuristic decision making, multiple asynchronous processes, and prioritized transaction based command management of multiple schedules and resources.

The Kennedy Space Center Ground Resource Allocation has applied an artificial intelligence based (object oriented) integrated payload scheduling system developed by Harris Corporation named PHITS (Payload Handling Inventory Tracking System).[12] Shuttle experiments are represented in terms of objects (with defined task flows) which are semantically related based on mission goals. The integrated schedules are generated by evaluating object/attribute/values with user defined constraints on object interaction. Schedule conflict resolutions are performed by the system executive to achieve the desired on-orbit requirement goal.

A prototype Customer Requirements Identification System (CRIS)[13] expert system was developed for NASA-KSC Space Shuttle payload processing operations. Based on past knowledge, strawman requirements are generated and updated for mission unique requirements. The breadth of CRIS is limited to animal research experiments and addresses requirements for test and checkout, test support, facilities, equipment, supplies, and special requirements.

SYSTEM INTEGRATION

Data Bases. The compilation of diverse elements into realizable system goals is aided through the application of highly developed data bases. An example of such a data base is called Requirements Driven Development (RDD)[14] for systems engineering methodology. RDD was developed by Ascent Logic Corporation to provide linkage between requirements, functional flows, and system behavioral models on one data base. The tool builds an element/relationship/attribute data base to facilitate expression of a complex system through layers of decomposition to the root system behavioral model. This tool is superior to traditional system requirement tools because it provides relationships between requirements traceability/hierarchy and system behavioral design models.

Expert Systems. Expert systems have been successfully demonstrated in some system integration applications. The Software Project Manager (SPM)[15] has been prototyped at Lockheed Software Technology Center as a decision support system for project management. It uses semantic networks for the knowledge representation in the model and functions, rules, and procedures to deductively reason over the model. Modification of the SPM knowledge base allows easy global definition of new relations or attributes through the model inheritance features. This allows the creation of a complex conceptual model without the major modifications required to maintain traditional data base systems. SPM provides tracking of constraints and relations, organizations, schedules, phases, tasks, inputs, outputs, resources, deliverables, milestones, and budgets.

Lockheed has created the Shuttle Connector Analysis Network (SCAN)[16] to ensure flight readiness of the space shuttle electrical wiring. This expert system tracks and assesses on going electrical configuration changes and mirrors the thinking of system engineers as they trace orbiter wiring. The system employs distributed networks that allow paperless real time configuration status along with quick system response through electronic signatures. Shuttle wiring is tested for constraint and feasibility analysis and troubleshooting of failed components. The system went from concept to production in about two years.

Rockwell International Space Transportation Systems Division has developed the orbiter payload bay cabling expert system (EXCABL).[17] Operational usage of EXCABL has demonstrated a 20 to 1 increase in design productivity. This system solves the cable layout design automation problems associated with supplying electrical services to satellites and experiments for each mission payload manifest. The system couples the technical order selection expert system (EXTOL) and the cable selector expert system (EXSEL). Based on flight requirements documents, EXTOL generates a master listing containing all orbiter payload bay installation drawings. Based on payload unique interface control documents (ICDs), EXSEL selects cables to be used by EXCABL to form a standardized orbiter cable inventory.

Hybrid Simulations. New hybrid simulation technologies are merging conventional programming with artificial intelligence techniques such as expert systems. Automatic Routing Module (ARM)[18] is an expert system that couples algorithmic and mission trajectory planning activities for Air Launched Cruise Missile long range terrain following routing. ARM was developed by Systems Control Technology for the Joint Strategic Target Planning Staff. ARM generates quality candidate routes for missions described by a launch point and a target. The routes are optimal in terms of threat avoidance and adhere to routing constraints including vehicle performance and heuristics. These heuristics substituted hand generation of routes by specifying minimum fuel reserve, target damage, target avoidance, and route dispersions. Rather than relying on the human eye to sort through the possibilities, ARM with its three dimensional terrestrial models provides a deterministic approach to find the safest path through high danger areas. The resulting routes proved to be of superior quality than manually generated routes. The Robotic Air Vehicle Program (RAV)[19] has been developed by Texas Instruments. This program combines technologies as control theory and navigational terrain algorithms with cooperating expert systems that can plan, execute, and alter the mission scenario.

The Advanced Launch System Moduling (ALSYM) tool was developed by Martin Marietta to provide rapid integration of ALS design concepts in terms of cost, operability, reliability, maintainability, and performance. ALSYM is implemented in an object oriented simulation environment providing cost and schedule impact analysis along with system performance assessment.

REQUIREMENTS VERIFICATION

Expert Systems. Verification of compliance to system requirements may be automated through application of expert systems. An online and real time Launch Readiness Assessment System (LRAS)[20] Manager was developed by EG&G contractor for the space shuttle. The system compares current status against requirement categories of personnel training, certifications, qualifications, maintenance, system/facility and equipment validations, and mission spare parts. Constraints, concerns, deviations, and waivers are also accounted for. Detailed requirements are derived from master data base files that contain the milestones, responsible unit relationships, and requirement assessment categories. The master data base files are maintained by the LRAS manager with inputs from each responsible unit as new equipment/facility/systems requirements enter or leave the expert system. LRAS provides a closed loop system for EG&G to track and verify all its support requirements for real time launch readiness assessment.

The Advanced Launch System Program is envisioned to make extensive use of knowledge based expert systems for automated ground prelaunch processing.[21] The expert system could monitor the launch vehicle system to provide command and monitor capability. If an anomaly occurs, recursive logic could be used to sort through combinations of component states to determine the system responsible for the anomaly. Once the proper inference engine has been developed, it can be applied to a new system configuration by loading a new knowledge base. Therefore, software development and maintenance costs would be reduced to a fraction of the cost for conventional software. Similarly, the ALS program envisioned usage of expert systems for post flight data processing. The software would be designed to analyze all data and report anomalies along with identifying the source of the problem and the rationale used to find it. Only for cases where the software could not isolate the anomaly factors would manual data review be required.

Hybrid Systems. The coupling of expert system applications and neural networks is finding an attractive niche in requirement verification. Neural networks have excelled at pattern recognition techniques for time varying signals. Neural networks can learn to discriminate nominal patterns based on examples and training; thus, eliminating the need for elaborate models or probability functions. Neural networks have proven to be robust for nonlinear processes as opposed to traditional techniques (Kalman filtering) which work well in linear regions. A prelaunch expert system is considered for the Orbiter Maneuvering System.[22] The approach includes reducing ground processing by including expert built in test and artificial neural networks. Neural network applications have been proven successful in detecting anomalies of space shuttle main engine sensor data.[23]

SYSTEMS ENGINEERING CONCEPTUAL MODEL

Systems engineering methodology will face an increasing challenge to capture engineering design knowledge for future applications in order to maintain an organization's competitive edge. The goal for payload integration design is to minimize the amount of change from mission to mission; thereby, reducing paperwork, labor hours, and turnaround time which are a reflection of poor standardization and information. Expert system technologies offer successful solutions for design automation applications and recurring integration design problems due to mission unique requirements.

Expert systems need to be applied with a systems engineering global approach in order to streamline activities from the outset. A Titan launch system engineering methodology study showed that in each system state (design, design verification, pre-mission, mission, and post-mission) a logic network display could be developed to order and classify the technical parameters involved in the translation from general to specific functional group requirements.[24] This type of methodology can now be captured in the knowledge base and inference engine.

Expert systems can be introduced for rapid prototyping prior to mission design integration and for design status functions during integration as shown in Figure 3. Rapid prototyping involves identifying those mission unique requirements from interface control documents that often cause analysis reiteration during integration or those requirements whose values can be precisely determined from the outset through early analyses. By capturing the surface level relationships between certain basic requirements, the technical parameters they affect, and how these parameters gain fidelity as they propagate across the functional groups; rapid prototyping can occur. The expert system can confirm if the mission unique requirements appear reasonable or can suggest which functional groups/analyses present a problem in meeting the requirement. Only after this iteration would mission design integration begin with fewer cross functional design changes or requirements modifications. Similarly, expert systems can maintain a common knowledge platform of engineering analysis, test, and manufacturing requirements compliance during the integration cycle. The expert system would be the focal point for requirements, functional flows, and analyses results. It could provide a snapshot of current integration activities and requirement compliance values. The system could perform downstream analysis to locate the effects of requirements to downstream tasks, or upstream analysis to identify dependencies on upstream tasks affecting a particular requirement. The expert system could suggest courses of action.

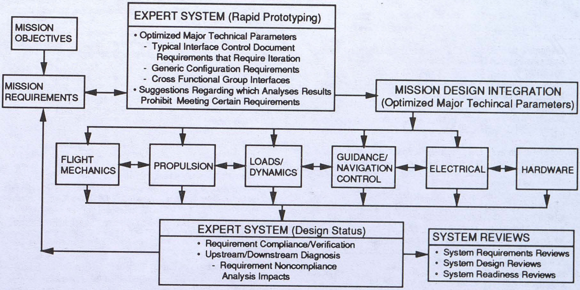

Figure 3. Payload Integration Design Cycle Conceptual Model

As technology develops, these applications will become easier to implement. PARAGON[25] representation, management, and manipulation system was developed by Ford Aerospace and Communications. PARAGON has a specific methodology for the construction and testing of expert systems that consists of a hybrid representation scheme integrating frames, semantic networks, classification hierarchies, blackboards, demons, transition networks, and rules. This allows concepts to be defined, their interrelationships specified, and their behavior described within a classification hierarchy. The knowledge domain is structured in dimensions (definition, composition, functional relationships, and sequential behavior). Information is propagated between concepts to varying degrees of knowledge description granularity. By defining how information is propagated through various relationships and constraining how concepts can be related, PARAGON achieves a uniform and consistent set of relations that moves the level of reasoning from a shallow to a deep level.

Application of expert systems offers the opportunity for analysis and interpretation of large amounts of data. Expert systems can keep track of the dynamics of systems engineering activities and can detect system anomalies with the proper inference mechanism.

CONCLUSION

Expert systems can enhance an organization's quality through closed loop reverse engineering applications. For example, in order to identify the sensor data output based on an input of a specific component failure, system simulations with numerical models can be used. However, the reverse problem of identifying the failed component, given certain sensor behavior, cannot be solved soley through numerical techniques to identify perhaps multiple anomalies. This complex diagnostic process, demanding hours of expert's discussions, can be approached by an expert system knowledge-based search capturing the interaction of system components.

Expert systems technology can serve a unifying function to the diverse engineering analyses and interactions. The application of these systems within the organization must be balanced withoptimism and skepticism. Expert systems must be placed in a setting where the organization's long term strategy is aligned with the expense and commitment necessary to modify existing organizational structure to accommodate technology.

Well understood mature systems could benefit from expert systems technology to streamline system processes. Such application could result in the difference between product life cycle decline or renewal. Expert systems software is a support tool applied to reasoning processes just as traditional software programs are applied to numerical processes. They offer a new realm of competitive technology applications and provide the mechanism to optimize system processes toward concurrent engineering.

REFERENCES

1. Crosby, Philip, Quality Improvement Through Defect Prevention, Philip Crosby Associates, Inc., 1985.
2. Winner, Robert, I.; Penel, James P.; Bertrand, Harold, E.; Slusarczuk, Marko, M. G.; The Role of Concurrent Engineering in Weapons Systems Acquisition, IDA Report R-338, December, 1988.
3. Becker, Shirley, A.; Hermer, Alan, R.; "Concurrent system Design with Box Structures", Thirteenth Annual International Computer Software and Applications Conference, IEEE, September, 1989.
4. DARPA Neural Network Study, AFCEA Press, 1988.
5. Slatter, Philip, E.; Building Expert Systems, Ellis Horwook Limited, 1987.
6. Bowerman, Robert, G.; Glover, David, E.; Putting Expert Systems into Practice, Van Nostrand Reinhold Company, Inc., 1988.
7. Edmunds, Robert, A.; The Prentice Hall Guide to Expert Systems, Prentice Hall Inc., 1988.
8. Gevarter, William B., "Expert System Terms and Building Tools", Machine Intelligence and Autonomy for Aerospace Systems, AIAA, Inc., 1988.
9. Oberright, John; McCain, Harry; Whitman, Ruth; "Space Station Flight Telerobotic Services Functional Requirements Development", Space Station Automation, SPIE, November, 1987.
10. Crawford, James, E.; Parnell, Gregory S.; "LRS-II: A Specialized Knowledge System for Launch Resource Scheduling", Twenty-Fifth Space Conference Proceedings, Canaveral Council of Technical Societies, April, 1988.
11. Britt, Daniel; Geoffrory, Amy; Schaeffer, Philip; "Scheduling Spacecraft Operations", Third Conference of Artificial Intelligence for Space Applications, NASA, November, 1987.
12. Ihrie, D.; Castillo, D.; McDaniel, M.; Tilley, R.; "An AI Approach for scheduling Space Station Payloads at Kennedy Space Center", Third Conference of Artificial Intelligence for Space Applications, NASA, November, 1987.
13. Ragusa, James, M.; Gonzales, Avelmo, J.; "Expert System Prototype Developments for NASA-KSC Business and Engineering Applications", Twenty-Fifth Space Conference Proceedings, Canaveral Council of Technical Societies, April, 1988.
14. Requirements Driven Development An Overview, Ascent Logic Corporation, 1989.
15. Bimson, Kent, D.; Boehms, Burris, Linda; "Conceptual Model Based Reasoning for Knowledge based software project management", Proceedings of the Twenty-First Hawaii International Conference System Sciences, IEEE, 1988.
16. Wilkinson, John; Trilley, Jim; "Automated Analysis of Shuttle wiring using SCAN", Twenty-Fifth Space Congress Proceedings, Canaveral Council of Technical Societies, 1988.
17. Morris, Keith, "Expert System Solutions to Space Shuttle Payload Integration Design Automation Problems", Twenty-Seventh AIAA Aerospace Sciences Meeting, AIAA, 1989.
18. Malin, Janice, "Automatic Routing Module", First Annual Workshop of Space Operation Automation and Robotics, NASA, October, 1987.
19. McNulty, Christa; Graham, Joyce; Roewer, Paul; "Robotic Air Vehicle", First Annual Workshop of Space Operation Automation and Robotics, NASA, October, 1987.
20. Geaslen, W. D., "EG&G Florida Inc., KSC Base Operations Contractor Launch Readiness Assessment System", Twenty-Fifth Space Congress Preceedings, Canaveral Council of Technical Societies, 1988.
21. Hickan, Robert, "ALS Automated Test and Checkout Philosophy", Proceedings of Aerospace Testing Seminar, Institute of Environmental Sciences, October, 1988.
22. Eagle, James; Owen, Charles; Colmearez, Luis; "Implementation of Expert System/AI technology for Reducing Ground Test in Present and Future Launch Systems", Aerospace Sciences Meeting, January, 1991.
23. Whitehead, Bruce; Ferher, Harry; Moonis, Ali; "Neural Network Approach to Space Shuttle Main Engine Heath Monitoring", Joint Propulsion Conference, AIAA, July, 1990.
24. Nelson, James, Methodology for the Systems Engineering Process, Contract NAS8-27567, Martin Marietta Corporation, March, 1972.
25. Ferguson, Jay C.; Wagner, Robert E.; "Beyond Rules: The Next Generation of Expert Systems", First Annual Workshop of Space Operation Automation and Robotics, NASA, October, 1987.